改进引力搜索最小二乘支持向量机交通流预测 *

徐钦帅 [a, b]，何　庆 [a, b†]，魏康园 [a, b]

(贵州大学 a. 大数据与信息工程学院; b. 贵州省公共大数据重点实验室，贵阳 550025)

摘　要：在智能交通系统领域中，准确的交通流量预测发挥着重要作用。为了提高基于最小二乘支持向量机的交通流预测模型的精度，提出一种新的改进引力搜索算法（TCK-AGSA）对其进行参数寻优。首先，基于 Tent 映射改进 Kbest 函数，使算法具有跳出局部最优的机制；然后，引入全局最优引导策略，使粒子加速朝向最优解移动；接着，将进化度因子和聚合度因子引入速度更新权重系数，使算法具有较强的自适应能力。针对 12 个基准函数的仿真结果表明，TCK-AGSA 的性能优于 GSA 及其改进算法。最后，建立基于 TCK-AGSA 寻优的最小二乘支持向量机模型，并选取 2016 年贵州省高速公路真实交通流数据进行预测实验，结果表明该模型具有更好的预测精度、鲁棒性和泛化能力。

关键词：引力搜索算法；混沌优化算法；自适应权重系数；最小二乘支持向量机；交通流预测

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2018.07.0383

Traffic flow forecasting using least squares support vector machine optimized by modified gravitational search algorithm

Xu Qinshuai[a, b], He Qing[a, b†], Wei Kangyuan[a, b]

(*a. College of Big Data & Information Engineering*, *b. Guizhou Provincial Key Laboratory of Public Big Data Guizhou University*, *Guiyang* 550025, *China*)

Abstract: The accuracy of traffic flow forecasting plays an important role in the field of Intelligent Transportation Systems. In order to improve the accuracy of traffic flow forecasting model based on Least Squares Support Vector Machine, this paper proposed a novel modified gravitational search algorithm (TCK-AGSA) for parameters optimization. Firstly, this paper improved the Kbest function based on Tent map, so that the algorithm has a mechanism to jump out of local optimum. Then, by introducing the guidance of global optimal to accelerate the movement of agents towards optimal solution. Furthermore, it introduced the evolutionary factor and converge factor into the weighted coefficient of agent's velocity to make the algorithm more adaptive. The simulation results for 12 benchmark functions show that the performance of TCK-AGSA is better than GSA and its variants. Finally, this paper proposed a LSSVM model optimized by TCK-AGSA, and selected the 2016 actual traffic flow data of Guizhou Expressway for experiment. The results show that the proposed model has better prediction accuracy, robustness, and generalization capability.

Key words: gravitational search algorithm; chaos optimization algorithm; adaptive weighted coefficient; least squares support vector machine; traffic flow forecasting

0　引言

随着经济的飞速发展和机动车保有量的持续上升，交通道路网络承受着越来越多的压力，存在交通拥堵、交通事故和尾气污染等问题。智能交通系统[1]（intelligent traffic system，ITS）的交通诱导控制、交通状况监控等技术的快速发展，能够针对当前交通痛点问题提供有效的解决方案。交通流预测作为 ITS 中交通诱导的重要依据，近几十年来的发展引起了广泛关注[2]。实现精准的交通流量预测，对于精确分析道路交通状况、改善交通网络规划和优化交通控制策略具有重要意义。

为了提高交通流预测的精度，国内外许多学者致力于交通流量数据的分析，并提出了多种预测模型。总的来看，已有文

收稿日期：2018-07-20；**修回日期：**2018-08-29　**基金项目：**贵州省科技计划项目重大专项资助项目（黔科合重大专项字［2018］3002）；贵州省公共大数据重点实验室开放课题（2017BDKFJJ004）；贵州省教育厅青年科技人才成长项目（黔科合 KY 字［2016］124）；贵州大学培育项目（黔科合平台人才［2017］5788）

作者简介：徐钦帅（1994-），男，山东滕州人，硕士研究生，主要研究方向为机器学习、智能计算（qsxu94@foxmail.com）；何庆（1982-），男（通讯作者），贵州贵阳人，副教授，博士，主要研究方向为大数据应用、人工智能；魏康园（1991-），女，陕西渭南人，硕士研究生，主要研究方向为数据挖掘、智能计算.

献所提出的交通流预测方法一般可以概括为两类：

a）基于经典统计学的预测模型。该模型通过对交通流数据进行回归和参数优化，从而实现交通流数据的拟合预测。例如，Kumar 等人 [3]在自回归移动平均模型基础上，考虑车流量受不同季节的影响，提出了一种基于季节的 SARIMA 模型，有效地解决了基本 ARIMA 模型对于交通流预测的适用性问题；随后，Kumar[4]于 2017 年提出了基于 Kalman 滤波技术的交通流预测模型，克服了 SARIMA 模型依赖于大量数据进行开发的缺陷。

b）基于数据驱动的预测模型。该模型通过智能计算方法挖掘历史交通流数据的演化趋势，从而预测未来数据的变化趋势。例如，在深度学习的应用方面，Lv 等人[5]提出了一种堆叠自动编码 SAE 混合深度学习的预测模型，有效地提高了交通流预测精度；Polson 等人[6]提出了一种结合使用 L1 正则化和 tanh 层序列进行拟合的线性深度学习模型，解决了由于交通状态之间的转换而产生的急剧非线性变化趋势的预测难题。在支持向量机（support vector machine，SVM）的应用方面，Hu 等人[7]提出了利用粒子群优化算法（particle swarm optimization，PSO）寻优支持向量回归参数的 PSO-SVR 交通流预测模型，并通过引入交通流历史动量降低噪声对预测精度的影响，获得了较优的预测性能；康军等人[8]通过简化 Lagrange 乘子求解过程，提出了一种基于 LSSVM 的在线式交通流预测方法，并在保证较高预测精度的情况下，有效缩短了在线模型的更新时间；Shang 等人[9]提出了基于相空间重构和组合核函数的最小二乘支持向量机（least squares support vector machine，LSSVM）预测模型，并利用 PSO 算法对其进行参数优化，具有较强的预测能力和鲁棒性；Cong 等人[10]提出了利用果蝇优化算法寻优参数的 FOA-LSSVM 预测模型，经实验验证该模型的预测效果明显优于 PSO-LSSVM 模型和径向基神经网络（RBFNN）模型。通过对上述预测模型的分析，发现基于数据驱动的预测模型对于非线性和不确定性的交通流预测更具有适用性。

相比于其他基于数据驱动的预测模型，LSSVM[11]模型在交通流预测方面表现出较好的性能，且由于其训练时间短和泛化能力强的优点，正被广泛应用于多种领域，如模型质量评估[12]、货运量预测[13]、短时风能预测[14]、PM 2.5 浓度预测[15]、汽轮机热耗率预测[16]、短时风速预测[17]和软件故障检测[18]等。在 LSSVM 模型中，正则化参数 C 和核宽度 σ 的选取至关重要，决定着模型的复杂度、预测精度及泛化能力。已有研究中，粒子群优化算法[9]、布谷鸟算法[15]、蚁群算法[17]等群智能优化算法被用于寻优 LSSVM 参数，且取得了较好的结果。但目前关于 LSSVM 最优参数组合选取方法的研究，仍处于探索阶段，且其预测准确性尚未达到理想精度。

引力搜索算法[19]（gravitational search algorithm，GSA）由 Rashedi 教授等于 2009 年提出，该算法源于物理学中的万有引力定律及牛顿第二定律。现有研究已证明[20]，GSA 算法的收敛性能明显优于 PSO 算法、遗传算法等算法。但是，基本 GSA 算法仍存在早熟收敛、易陷入局部最优和缺少记忆功能等问题。基于此，国内外学者对 GSA 算法进行了很多有效的改进，如 Darzi 等人[21]将 PSO 算法中个体局部最优值概念引入 GSA 算法中，弥补了算法关于记忆性的空缺；Zhang 等人[22]提出了基于自适应引力系数衰减因子、精英记忆引导和变异操作的混合改进策略，较好地提高了 GSA 算法的收敛性能；郭洁皓等人[23]提出了一种基于信息熵模型和动态选择权重的改进 GSA 算法，提高了算法的收敛速度与寻优效率。尽管现有改进策略各有优势，但对于平衡算法的探索与开发能力及改善算法跳出局部最优能力仍是难点。

因此，基于上述 GSA 算法的缺陷，提出一种基于 Tent 映射混沌 Kbest 的自适应 GSA 算法（tent map based chaotic kbest-adaptive gravitational search algorithm，TCK-AGSA）。首先，结合混沌优化算法中的 Tent 映射对 GSA 算法的精英粒子集合 Kbest 进行改进，使其呈非线性递减变化趋势，以便算法能够跳出局部最优；然后，在粒子速度更新式中加入全局最优值引导机制，并引入综合种群进化程度和聚合程度的自适应权重系数，在加速算法对求解空间搜索的同时，使得算法能够自适应性调整。实验选取 12 个基准测试函数进行仿真对比，检验改进策略的有效性。最后，利用改进算法对 LSSVM 的参数组合进行迭代寻优，构建 TCK-AGSA-LSSVM 交通流预测模型，并应用于真实交通流数据集，通过预测均方误差（MSE）等评价指标检验该模型对于实际交通流预测应用的优越性。

1 相关工作

1.1 最小二乘支持向量机

最小二乘支持向量机（LSSVM）优化并扩展了支持向量机（SVM）的数学模型：用等式约束替代 SVM 中的不等式约束，采用基于 L2 范数的误差函数作为损失函数，将 SVM 的 QP 求解问题通过最优条件转换为线性方程组求解问题。其基本模型理论如下：

给定一个包含 N 个样本的训练样本集 $D=\{x_i, y_i\}_{i=1}^{N}$，其中 x_i 为输入向量，$x_i \in \mathbf{R}^d$；y_i 为相应的输出向量，$y_i \in \mathbf{R}$。LSSVM 利用非线性映射函数 $\varphi(x)$ 将样本向量映射到高维空间中，构造线性回归函数为

$$y(x) = \boldsymbol{w}^{\mathrm{T}}\varphi(x) + b \tag{1}$$

其中：$\boldsymbol{w}$ 为权重矢量；b 为偏置。

依据结构风险最小化原则，利用 LSSVM 进行回归时的优化问题为

$$\min J(\boldsymbol{w}, \xi) = \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + \frac{1}{2}C\sum_{i=1}^{N}\xi_i^2 \tag{2}$$

$$s.t.\quad y_i = w^{\mathrm{T}} j(x_i) + b + x_i\ . \tag{3}$$

其中：C 为正则化参数；ξ 为误差变量；$i \in \{1, \cdots, N\}$。

为求解式（2），利用拉格朗日乘子法求其对偶问题，构造拉格朗日函数为

$$L(\boldsymbol{w}, b, \xi, \alpha) = J(\boldsymbol{w}, \xi) - \sum_{i=1}^{N}\alpha_i(\boldsymbol{w}^{\mathrm{T}}\varphi(x_i) + b + \xi_i - y_i) \tag{4}$$

其中：α_i 为拉格朗日乘子。

根据 KKT 条件，分别求出 $L(\boldsymbol{w},b,\xi,\alpha)$ 对 $\boldsymbol{w},b,\xi,\alpha$ 的偏导，得出式（2）的最优条件为

$$\begin{cases}\dfrac{\partial L}{\partial \boldsymbol{w}}=0\rightarrow \boldsymbol{w}=\sum_{i=1}^{N}\alpha_i\varphi(x_i)\\ \dfrac{\partial L}{\partial b}=0\rightarrow \sum_{i=1}^{N}\alpha_i=0\\ \dfrac{\partial L}{\partial \xi_i}=0\rightarrow \alpha_i=C\xi_i\\ \dfrac{\partial L}{\partial \alpha_i}=0\rightarrow \boldsymbol{w}^{\mathrm{T}}\varphi(x_i)+b+\xi_i-y_i=0\end{cases} \tag{5}$$

简化消去 $\boldsymbol{w}$ 和 ξ_i 可得出

$$\begin{bmatrix}0 & \boldsymbol{E}^{\mathrm{T}}\\ \boldsymbol{E} & \boldsymbol{K}+C^{-1}\boldsymbol{I}\end{bmatrix}\begin{bmatrix}b\\ \boldsymbol{\alpha}\end{bmatrix}=\begin{bmatrix}0\\ \boldsymbol{y}\end{bmatrix} \tag{6}$$

其中：$\boldsymbol{y}=[y_1,y_2,\cdots y_N]^{\mathrm{T}}$；$\boldsymbol{E}$ 为全 1 序列；$\boldsymbol{I}$ 为相应维数的单位矩阵；$\boldsymbol{K}$ 为核函数矩阵；

$$\boldsymbol{K}_{ij}=\varphi(x_i)^{\mathrm{T}}\varphi(x_i)=k(x_i,x_j) \tag{7}$$

其中：$k(x_i,x_j)$ 为核函数。令 $\boldsymbol{Q}=(\boldsymbol{K}+C^{-1}\boldsymbol{I})$，代入式（6）中求解可得

$$b=\frac{\boldsymbol{E}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{y}}{\boldsymbol{E}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{E}} \tag{8}$$

$$\boldsymbol{\alpha}=\boldsymbol{Q}(\boldsymbol{y}-\frac{\boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{y}}{\boldsymbol{E}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{E}}) \tag{9}$$

最终的 LSSVM 模型回归函数表示为

$$y(x)=\boldsymbol{w}^{\mathrm{T}}\varphi(x)+b=\sum_{i=1}^{N}\alpha_i k(x_i,x_j)+b \tag{10}$$

针对核函数的选取，采用参数相对较少、适应能力较强的高斯核函数（Radial Basis Function，RBF）作为 LSSVM 的核函数，定义为

$$k(x_i,x_j)=\exp(-\frac{\left\|x_i-x_j\right\|^2}{2\sigma^2}) \tag{11}$$

其中：σ 为高斯核的带宽（width）。

通过分析 LSSVM 回归理论可知，正则化参数 C 和高斯核带宽 σ 作为 LSSVM 回归优化问题的重要参数组合，直接影响着模型的学习与泛化能力。1）正则化参数 C 能够平衡模型的结构风险与经验风险，其取值越大，经验风险越小，易造成过拟合；反之，则降低了模型复杂度，易造成欠拟合。2）高斯核带宽 σ 取值越大，核函数越平滑，易造成欠拟合；反之，核函数越陡峭，易造成过拟合。因此，针对不同优化问题选取相适应的模型参数组合尤为关键，文中采用改进的引力搜索算法对 LSSVM 参数组合进行寻优。

1.2　引力搜索算法

在引力搜索算法（GSA）的迭代进化过程中，将每个粒子视为遵循牛顿第二定律，在搜索空间中进行无阻力运动的有质量的物体，其运动会受到解空间中其他粒子万有引力的影响。基本 GSA 算法表述如下：

由粒子 X_i 构成的规模为 N 的种群，位于一个 D 维搜索空间中，粒子 X_i 的位置可表示为 $X_i=(x_i^1,x_i^2,\cdots,x_i^d,\cdots,x_i^D)$，其中 x_i^d 表示第 i 个粒子在第 d 维上的位置。在第 t 次迭代进化，粒子 i 的惯性质量 $M_i(t)$ 依据其适应度值计算，计算式为

$$mass_i(t)=\begin{cases}\dfrac{fit_i(t)-worst(t)}{best(t)-worst(t)} & if\ best(t)\neq worst(t)\\ 1 & otherwise\end{cases} \tag{12}$$

$$M_i(t)=\frac{mass_i(t)}{\sum_{j=1}^{N}mass_j(t)}. \tag{13}$$

其中：$fit_i(t)$ 表示为算法进化至第 t 代时，粒子 i 的适应度值，$i\in\{1,\cdots,N\}$。针对最小值优化问题，最优适应度 $best(t)$ 和最差适应度 $worst(t)$ 分别为

$$best(t)=\min_{j\in\{1,\cdots,N\}} fit_j(t) \tag{14}$$

$$worst(t)=\max_{j\in\{1,\cdots,N\}} fit_j(t). \tag{15}$$

反之，即可用于最大值优化问题。

在第 t 次进化迭代，粒子 i 与粒子 j 在第 d 维的相互吸引力为

$$F_{ij}^d(t)=G(t)\frac{M_i(t)\times M_j(t)}{R_{ij}(t)+\varepsilon}(x_j^d(t)-x_i^d(t)) \tag{16}$$

其中：M_i 与 M_j 分别表示为粒子 i 与粒子 j 的惯性质量；ε 是一个很小的常量；$G(t)$ 为第 t 代的万有引力系数；$R_{ij}(t)$ 为粒子 i 与粒子 j 之间的欧氏距离，计算式分别为

$$G(t)=G_0\times e^{(-\alpha\frac{t}{T})} \tag{17}$$

$$R_{ij}(t)=\left\|X_i(t),X_j(t)\right\|_2 \tag{18}$$

其中：G_0 为初始引力常量；α 为引力系数变化系数；T 为最大迭代次数。

为了使算法具有随机性，在第 d 维上粒子 i 所受合力为

$$F_i^d=\sum_{j=1,j\neq i}^{Kbest} rand_i\times F_{ij}^d(t) \tag{19}$$

其中：Kbest 初始值为 $K_0=N$，随着时间推移，Kbest 逐渐减小为 1；$rand_i$ 为区间 $[0,1]$ 的随机数；F_{ij}^d 的定义如式（16）所示。

算法进化至第 t 代时，第 d 维上粒子 i 的加速度定义为

$$a_i^d(t)=\frac{F_i^d(t)}{M_i(t)} \tag{20}$$

在 GSA 算法进化过程中，对于每一次迭代，根据式（21）（22）分别更新粒子 i 的速度与位置，分别定义为

$$v_i^d(t+1)=rand_j\times v_i^d(t)+a_i^d(t) \tag{21}$$

$$x_i^d(t+1)=x_i^d(t)+v_i^d(t+1) \tag{22}$$

其中：$rand_j$ 为区间 $[0,1]$ 的随机数。

分析基本算法可知：a）随着算法迭代次数增加，种群的多样性会逐渐减小，从而可能丢失质量较小却保持着较好进化趋势的粒子，致使算法早熟收敛和陷入局部最优值；b）根据速度与位置更新式（21）（22）可知，算法在迭代进化过程中仅利用当前粒子位置信息引导算法搜索最优解，缺少群体记忆功能与群体信息共享机制，导致可能失去先前迭代的最优结果，并加速粒子朝向新的、可能较劣的位置移动，失去最优搜索轨迹。

因此，针对 GSA 算法缺少跳出局部最优机制和群体记忆功能等缺陷，对其进行改进以提高其搜索寻优精度，进而应用于 LSSVM 模型的关键参数寻优，实现更加精确的交通流预测。

2　改进引力搜索算法设计

2.1　基于 Tent 映射的混沌 Kbest

在 GSA 算法中，Kbest 为具有较优适应度值和较大质量的精英粒子集合。随着算法的迭代进化，Kbest 函数值的减少能够提高 GSA 算法的计算效率。已知 GSA 算法中，第 t 次迭代时 Kbest 定义为

$$Kbest(t)=final_per+(\frac{1-t}{T})\times(100-final_per) \tag{23}$$

其中：$final_per$ 为对其他粒子产生作用力的粒子百分比。

由式（23）可知，Kbest 是一种随着迭代进化而线性递减的函数。针对算法易陷入局部最优值的缺陷，利用混沌映射的随机性和遍历性，对 Kbest 进行混沌优化使其非线性递减，从而使得算法具有跳出局部最优的能力。文献[24]提出的 CKGSA 算法将 Logistic 映射应用于 GSA 算法中进行改进，对算法的搜索性能有所提高，但是对部分单模态函数和高维多模态函数的收敛精度仍低于基本 GSA 算法，这是由于 Logistic 映射具有遍历不均匀性，其在 $[0,0.1]$ 和 $[0.9,1]$ 内取值概率较高，从而影响算法的搜索效率和收敛精度。

在混沌优化算法中，Tent 映射为一维的分段线性映射，其混沌序列分布均匀，相比于 Logistic 映射迭代速度更快，有更好的遍历均匀性与更高的寻优效率。Tent 映射的数学表达式定义为

$$x(t+1)=\begin{cases}2x(t) & 0\le x(t)\le 0.5\\ 2(1-x(t)) & 0.5<x(t)\le 1\end{cases} \tag{24}$$

其中：$x(t)\in[0,1]$ 为第 t 次迭代时的混沌数。

因此，本文利用 Tent 映射混沌机制对 GSA 算法中的 Kbest 函数进行改进，使其呈非线性递减趋势。基于 Tent 映射的混沌 Kbest 函数定义为

$$Kbest(t)=final_per\times x(t)+(N-final_per)\times(\frac{T-t}{T}) \tag{25}$$

其中：N 为种群规模；T 为算法最大迭代次数；$final_per$ 为对其他粒子产生作用力的粒子百分比。

2.2　自适应权重全局最优引导速度更新

针对 GSA 算法缺少记忆功能和最优值振荡的不足，将 GSA 算法速度更新式（21）改进为

$$\begin{aligned}v_i^d(t+1)=\omega\cdot v_i^d(t)+c_1\cdot rand_i\cdot a_i^d(t)+\\ c_2\cdot rand_j\cdot(gbest^d(t)-x_i^d(t))\end{aligned} \tag{26}$$

其中：ω 为速度权重系数；c_1,c_2 分别为粒子个体和群体记忆学习因子；$rand_i$ 和 $rand_j$ 为 $[0,1]$ 的随机数；$gbest^d(t)$ 为第 t 代群体最优解位置。

在速度更新式（26）中，速度权重系数 ω 对算法的寻优性能发挥重要作用。当权重系数 ω 较大时，算法具有较强的全局探索能力；反之，则利于算法的局部开发收敛。因此，本文进一步将自适应权重思想引入 GSA 算法，综合当代群体进化度和聚合度对权重系数 ω 进行自适应调整，以提高算法的寻优能力。

首先，在算法进化至第 t 代时，将粒子 i 的适应度值定义为 $f(x_i(t))$，粒子 i 的个体最优适应度值为 $f(pbest_i(t))$，群体最优适应度值为 $f(gbest(t))$。

然后，定义 GSA 算法的进化度因子 $evo_i(t)$ 为

$$\begin{aligned}evo_i(t)=a_1\cdot\frac{f(pbest_i(t-1))}{f(pbest_i(t))}+a_2\cdot\frac{avef(pbest_i(t-1))}{avef(pbest_i(t))}+\\ a_3\cdot\frac{f(gbest(t-1))}{f(gbest(t))}\end{aligned} \tag{27}$$

其中：a_1,a_2,a_3 为区间 $[0,1]$ 的相关系数，其关系定义为 $a_1+a_2+a_3=1$，且由于进化度因子 $evo_i(t)$ 以群体寻优度最为关键，因此 $a_3>\max(a_1,a_2)$；$avef(pbest(t))$ 为第 t 代群体中粒子的个体最优平均值，定义为

$$avef(pbest(t))=\frac{\sum_{i=1}^{N}f(pbest_i(t))}{N} \tag{28}$$

同时，定义 GSA 算法的聚合度因子 $con(t)$ 为

$$con(t)=\frac{avef(x(t))}{avef(pbest(t))} \tag{29}$$

其中：$avef(x(t))$ 为第 t 代群体中所有粒子适应度的平均值，定义为

$$avef(x(t))=\frac{\sum_{i=1}^{N}f(x_i(t))}{N} \tag{30}$$

已知当进化度因子较大时，表明群体进化程度较差，需减小权重系数 ω 以增强算法的局部开发能力；而当聚合度因子较大时，表明群体聚合程度较高，需增大权重系数以增强算法全局探索能力。因此，GSA 算法进化至第 t 代时，粒子 i 的自适应权重系数定义为

$$\omega_i(t)=\omega-b_1\cdot evo_i(t)+b_2\cdot con(t) \tag{31}$$

其中：ω 为初始权重系数；b_1 和 b_2 分别为进化度因子和聚合度因子的平衡系数，$0<b_1,b_2<\omega$ 且 $b_1+b_2=1$。

综上所述，结合提出的改进速度更新式（26）和自适应权重系数式（31），将算法第 $t+1$ 代粒子 i 的速度更新式定义为

$$\begin{aligned}v_i^d(t+1)=(\omega-b_1\cdot evo_i(t)+b_2\cdot con(t))\cdot v_i^d(t)+\\ c_1\cdot rand_i\cdot a_i^d(t)+c_2\cdot rand_j\cdot(gbest^d(t)-x_i^d(t))\end{aligned} \tag{32}$$

3　仿真实验与结果分析

本文采用 MATLAB R2013a 进行仿真实验，运行环境为 64 位 Windows 7 操作系统。选取 12 个基准测试函数进行数值实验，并与基本 GSA 算法、基于 Logistic 映射混沌 Kbest 的 CKGSA 算法[24]和基于全局最优值引导的 GG-GSA 算法[25]进行比较。

为保证算法测试实验中的公正性，基于文献[24,25]的参数设置，将 GSA 算法和 TCK-AGSA 算法的种群规模 N 设为 50，最大迭代次数设为 1000，初始万有引力常量设为 100。其他参数设置如表 1 所示。

表 1　算法参数设置

算法	参数设置
GSA	α=20
CKGSA[24]	α=20；μ=4
GG-GSA[25]	α=30；$c_1 = 2-1.9(t/T)$；$c_2 = t/T$
TCK-AGSA	α=20；$c_1 = 2-1.9(t^3/T^3)$；$c_2 = t^3/T^3$ $a_1 = a_2 = 0.2$；$a_3 = 0.6$；$b_1 = b_2 = 0.5$

表 2 为仿真实验采用的 12 个基准测试函数，包含三种类别：F_1-F_4 为单模态函数，特别的，F_4 为间断阶梯函数；F_5-F_8 为高维多模态函数；F_9-F_{12} 为低维多模态函数。采用形态各异的测试函数能有效检验 TCK-AGSA 算法的收敛速度、收敛精度和多峰寻优能力等优化性能。同时，为了有效减小随机干扰对寻优结果的影响，每个测试函数均独立运行 30 次，并取平均实验结果进行比较。

表 3 记录了 GSA 算法、CKGSA[24]算法、GG-GSA[25]算法和 TCK-AGSA 算法在相同测试条件下，对于 12 个基准测试函数的寻优结果。其中：*Mean*、*Best* 和 *Std.Dev* 分别表示为算法独立运行 30 次的平均值、最优值和标准差。在表 3 中，CKGSA 算法寻优结果来自于文献[24]中的表 3，GG-GSA 算法对于基准测试函数的寻优结果来自于文献[25]中的表 2；“—”表示该算法在对应文献中没有提供相应数据；字体加粗标注表示不同算法寻优结果中的最优值，其他表类似。

图 1 给出了 GSA 算法和 TCK-AGSA 算法对于选取自不同类别的 6 个基准测试函数寻优的收敛过程对比图，以更直观的检验 TCK-AGSA 算法的收敛性能，其中：t 表示迭代次数；f 表示算法运行 30 次的平均适应度值。

表 2　测试函数

函数名	表达式	维度	范围	理论最优值
Schwefel 1.2	$F_1(x)=\sum_{i=1}^{Dim}\left(\sum_{j=1}^{i}x_j\right)^2$	30	[-100,100]	0
Rosenbrock	$F_2(x)=\sum_{i=1}^{Dim-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	[-30,30]	0
Schwefel 2.22	$F_3(x)=\sum_{i=1}^{Dim}\lvert x_i\rvert+\prod_{i=1}^{Dim}\lvert x_i\rvert$	30	[-10,10]	0
Step	$F_4(x)=\sum_{i=1}^{Dim}\left(\lfloor x_i+0.5\rfloor\right)^2$	30	[-100,100]	0
Ackley	$F_5(x)=-20\exp\left(-0.2\sqrt{\frac{1}{\mathrm{Dim}}\sum_{i=1}^{Dim}x_i^2}\right)-\exp\left(\frac{1}{Dim}\sum_{i=1}^{Dim}\cos\left(2\pi x_i\right)\right)+20+e$	30	[-32,32]	0
Schwefel 2.26	$F_6(x)=-\sum_{i=1}^{Dim}\left(x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)\right)$	30	[-500,500]	-12569.5
Penalized	$F_7(x)=0.1\left\{\sin^2\left(3\pi x_1\right)+\sum_{i=1}^{Dim}\left(x_i-1\right)^2\left[1+\sin^2\left(3\pi x_i+1\right)\right]+\left(x_d-1\right)^2\left[1+\sin^2\left(2\pi x_d\right)\right]\right\}$ $+\sum_{i=1}^{Dim}u\left(x_i,5,100,4\right)$	30	[-50,50]	0
Griewank	$F_8(x)=\frac{1}{4000}\sum_{i=1}^{Dim}x_i^2-\prod_{i=1}^{Dim}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600,600]	0
Shekel's Foxholes	$F_9(x)=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right]^{-1}$	2	[-65.536,65.536]	1
Kowalik	$F_{10}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_2+x_4}\right]^2$	4	[-5,5]	0.0003
Goldstein-Price	$F_{11}(x)=\left[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)\right]\times$ $\left[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\right]$	2	[-2,2]	3
Shekel's Family	$F_{12}(x)=-\sum_{i=1}^{5}\left[\left(X-a_i\right)\left(X-a_i\right)^T+c_i\right]$	4	[0,10]	−10.15

由表 3 可知，在寻优单模态函数问题 F_1-F_4 时，TCK-AGSA 算法对函数 F_1、F_2、F_3 的寻优效果明显优于其他算法。只是对于函数 F_1 寻优 30 次的最优值（*Best*）略差于 CKGSA 算法，但是相比于基本 GSA 算法和 GG-GSA 算法，TCK-AGSA 算法的收敛精度和稳定性有很大幅度的提高。在图 1（a）、（b）中，TCK-AGSA 算法的收敛进化曲线表明其在寻优单模态函数问题时，收敛速度和收敛精度较 GSA 算法都有所提高，尤其是优化 F_3 时较基本 GSA 算法提高了 7 个数量级。而对于函数 F_4，

四种算法均能取得其理论最优值，表明 GSA 算法的寻优机制能够很好地求解间断阶梯函数。

表 3　TCK-AGSA 与其他算法寻优测试函数结果对比

函数	算法	*Mean*	*Best*	*Std.Dev*
F_1	GSA	2.954E + 02	2.895E + 02	8.675E + 01
	CKGSA[24]	1.501E + 01	**8.037E - 01**	1.074E + 01
	GG-GSA[25]	2.274E + 02	8.695E + 01	8.740E + 01
	TCK-AGSA	**1.067E + 01**	6.504E + 00	**5.134E + 00**
F_2	GSA	2.857E + 01	2.608E + 01	1.846E + 01
	CKGSA[24]	2.402E + 01	2.363E + 01	2.148E - 01
	GG-GSA[25]	2.621E + 01	2.556E + 01	7.316E - 01
	TCK-AGSA	**2.395E + 01**	**2.305E + 01**	**2.086E - 01**
F_3	GSA	2.313E - 08	2.229E - 08	3.357E - 09
	CKGSA[24]	2.450E - 08	1.870E - 08	4.640E - 09
	GG-GSA[25]	1.260E - 10	3.510E - 12	2.337E - 10
	TCK-AGSA	**2.513E - 15**	**2.130E - 15**	**1.878E - 15**
F_4	GSA	0.000E + 00	0.000E + 00	0.000E + 00
	CKGSA[24]	0.000E + 00	0.000E + 00	0.000E + 00
	GG-GSA[25]	0.000E + 00	0.000E + 00	0.000E + 00
	TCK-AGSA	**0.000E + 00**	**0.000E + 00**	**0.000E + 00**
F_5	GSA	3.596E - 09	3.205E - 09	3.250E - 10
	CKGSA[24]	3.600E - 09	2.510E - 09	4.720E - 10
	GG-GSA[25]	**2.703E - 12**	**4.343E - 13**	**3.218E - 12**
	TCK-AGSA	1.725E - 10	1.469E - 10	3.186E - 11
F_6	GSA	-2.257E + 03	-2.716E + 03	5.159E + 02
	CKGSA[24]	—	—	—
	GG-GSA[25]	-2.852E + 03	-4.731E + 03	6.178E + 02
	TCK-AGSA	**-3.081E + 03**	**-7.220E + 03**	**4.363E + 02**
F_7	GSA	1.500E – 03	1.072E - 18	3.800E - 03
	CKGSA[24]	1.500E - 03	8.500E - 19	3.800E - 03
	GG-GSA[25]	7.325E - 04	2.838E - 26	2.788E - 03
	TCK-AGSA	**2.084E - 04**	**1.824E - 26**	**9.086E - 04**
F_8	GSA	4.453E + 00	2.057E + 00	2.846E + 00
	CKGSA[24]	—	—	—
	GG-GSA[25]	1.170E + 00	3.946E - 02	8.023E - 01
	TCK-AGSA	**1.500E - 03**	**0.000E + 00**	**3.400E - 03**
F_9	GSA	3.467E + 00	1.992E + 00	2.794E + 00
	CKGSA[24]	3.986E + 00	9.980E - 01	2.464E + 00
	GG-GSA[25]	2.241E + 00	9.980E - 01	9.922E - 01
	TCK-AGSA	**1.396E + 00**	**9.980E - 01**	**5.445E - 01**
F_{10}	GSA	1.988E - 03	2.761E - 03	3.203E - 04
	CKGSA[24]	1.100E - 03	7.000E - 04	6.000E - 04
	GG-GSA[25]	7.119E - 04	5.362E - 04	5.918E - 05
	TCK-AGSA	**6.237E - 04**	**4.396E - 04**	**4.514E - 05**
F_{11}	GSA	3.000E + 00	3.000E + 00	8.308E - 15
	CKGSA[24]	3.000E + 00	3.000E + 00	2.340E - 15
	GG-GSA[25]	3.000E + 00	3.000E + 00	2.566E - 15
	TCK-AGSA	**3.000E + 00**	**3.000E + 00**	**7.364E - 16**
F_{12}	GSA	-5.706E + 00	-1.015E + 01	4.060E + 00
	CKGSA[24]	-7.926E + 00	-1.015E + 01	3.325E + 00
	GG-GSA[25]	-7.232E + 00	-1.015E + 01	3.347E + 00
	TCK-AGSA	**-8.740E + 00**	**-1.015E + 01**	**2.904E + 00**

在寻优高维多模态函数问题 F_5-F_8 时，对于函数 F_6、F_7、F_8，TCK-AGSA 算法寻优效果均明显高于其他算法，只是对于函数 F_5 的寻优精度低于 GG-GSA 算法。其中，函数 F_8 为 Griewank 函数，是一种典型的非线性多模态函数，具有广泛的寻优搜索空间和众多局部极值点，但是由表 3 和图 1（d）可以看出，TCK-AGSA 算法对于复杂函数 F_8 的寻优效果尤为显著，且取得函数理论最优值，这表明该算法基于 Tent 映射的混沌 Kbest 非线性递减策略，能够使得算法跳出局部最优值，进而有效收敛于全局最优值邻域。

与高维多模态函数相比，低维多模态函数 F_9-F_{12} 具有相对较少的局部极值点，能够用于检测算法的鲁棒性。由表 3 和图 1（e）（f）可知，TCK-AGSA 算法在收敛精度和收敛速度方面均优于其他算法，且三种改进算法对于 F_{11} 和 F_{12} 均分别取得了其理论最优值，其中 TCK-AGSA 算法寻优函数的平均值（*mean*）和标准差（*std.dev*）明显优于其他算法，表明其具有较好的鲁棒性。而由 GSA、CKGSA 算法的收敛效果较差可知，缺少记忆功能的引力搜索算法提高收敛性能的余地是有限的。

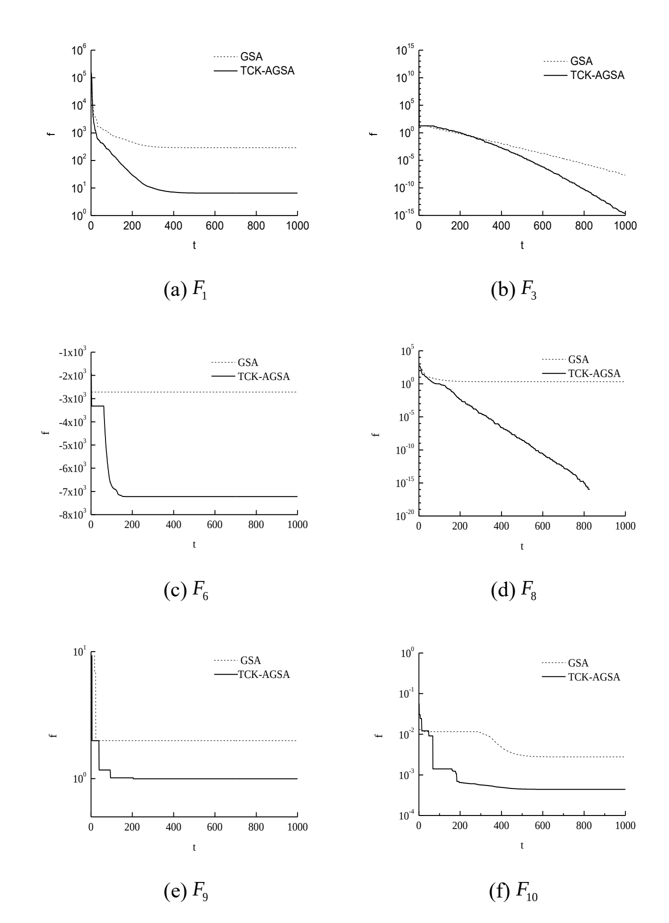


图 1　TCK-AGSA 与 GSA 寻优测试函数的收敛曲线图

综上所述，相比于基本 GSA 算法、CKGSA[24]算法和 GG-GSA[25]算法，TCK-AGSA 算法能够有效的处理复杂函数寻优问题，在单模态函数和多模态函数问题上均取得了较好的效果，收敛精度、收敛速度和鲁棒性均有所提高，验证了改进算法的非线性递减 Kbest 策略和自适应权重全局最优引导速度更新方式的优越性。

4 交通流预测应用

4.1 数据基础及预处理

本文采用的数据集为贵州省高速公路交通流量数据集，采集地点为尾号 D534 交通观测站，采集时间为 2016 年 11 月。该数据集包含不同车型的车流量、车速、车道号和平均车头间距等 30 个维度的数据。通过数据清洗、降维和统计，选取 2016 年 11 月 7 日至 11 月 11 日共计 5 天的交通流数据，包含 11 号、12 号、31 号和 32 号车道的交通流量，采集时间间隔为 15 分钟，每天 24 小时共计采集 96 条数据。该时间段数据为一周内工作日的交通流量，具有相对明显的周期性，将 11 月 7 日至 11 月 10 日（周一至周四）的交通流量作为训练集用于模型的学习，11 月 11 日（周五）的交通流量作为测试集用于模型的预测效果验证。

针对真实交通流数据的不稳定性、变化幅度较大的问题，根据式（33）对训练样本集进行归一化处理，以提升 LSSVM 模型的训练效率，降低训练复杂度，减少训练时间。归一化方法定义如下：

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{33}$$

其中：x 为交通流真实数据值；x' 为归一化后的交通流数据值；$x_{\max}$ 和 $x_{\min}$ 分别为交通流数据的最大车流量值和最小车流量值。

4.2 TCK-AGSA 算法优化 LSSVM 模型的交通流预测

根据交通流预测应用问题，建立 LSSVM 预测模型，并利用 TCK-AGSA 算法对其关键参数组合 C 和 σ 进行优化，提高该模型的学习效率，从而实现更精准的预测。模型输入为 TCK-AGSA 算法的初始参数，以及真实交通流量数据；输出为优化后参数组合 C 和 σ 的取值，以及交通流量预测结果。TCK-AGSA-LSSVM 预测模型实现流程如图 2 所示。

该模型具体实现步骤如下：

a)初始化 TCK-AGSA 算法的粒子数目 N、最大迭代次数 T 等参数，并随机部署粒子位置；初始化 LSSVM 参数，确定参数 C 和 σ 的取值范围，将其取值映射为 TCK-AGSA 算法的粒子位置坐标。

b)选取预测值与真实值的均方误差（MSE）作为算法寻优的适应度函数，如式（34）所示。基于 LSSVM 参数与引力搜索粒子位置相互映射关系，计算其适应度值。

c)根据式（12）和（13）计算更新惯性质量 $M_i(t)$，同时根据式（17）更新万有引力系数 $G(t)$。

d)根据式（16）计算粒子间的相互吸引力 $F_{ij}(t)$。

e)根据式（25）计算基于 Tent 映射的混沌 Kbest，并根据式（19）计算粒子在不同方向上所受合力。

f)根据式（20）和（31）分别计算粒子的加速度和速度权重系数。

g)根据式（32）更新粒子速度 v，然后根据式（22）更新粒子位置 x。

h)判断是否已达到算法终止条件，若是则执行 Step9；否则跳至 b)，循环迭代。

i)根据粒子位置与 LSSVM 参数的映射关系，确定全局最优粒子的位置 x，即为 LSSVM 模型的最优参数寻优组合 C 和 σ，并利用最优参数组合建立 TCK-AGSA-LSSVM 预测模型。

j)利用优化后的模型对交通流数据进行训练学习，最终输出交通流预测值，并通过计算测试样本值与预测值的误差来评价模型预测性能。

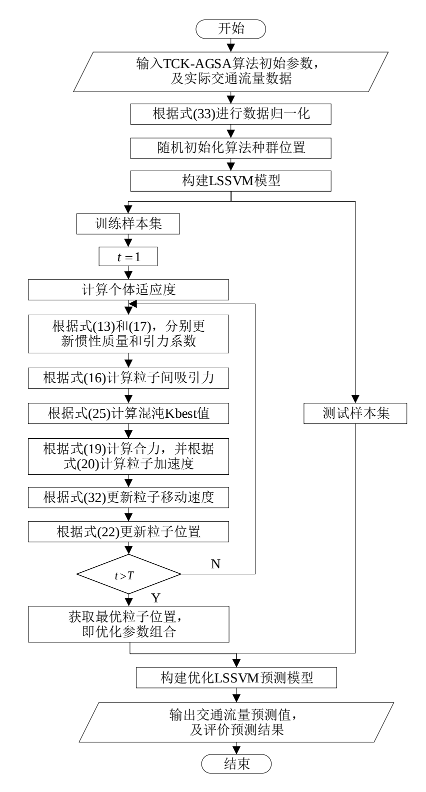


图 2　TCK-AGSA 优化 LSSVM 预测模型实现流程

4.3 参比模型与评价指标

为了充分验证 TCK-AGSA 算法寻优 LSSVM 模型对于交通流预测的有效性和优越性，选取基本 LSSVM 模型、文献[10]提出的 FOA-LSSVM 模型和文献[14]提出的 GSA-LSSVM 模型进行仿真实验，并与 TCK-AGSA-LSSVM 模型的预测结果进行对比分析。为了方便比较，将 FOA、GSA 和 TCK-AGSA 算法的种群规模均设为 $N=50$，最大迭代次数为 $T=100$，寻优维度 $D=2$。特别的，FOA 算法寻优步长设为 10；GSA 和 TCK-AGSA 算法的其他参数设置同表 1；基本 LSSVM 的正则化参数 $C=0.1$，核宽度 $\sigma=500$。

针对不同交通流预测模型的评价与比较，采用均方误差（MSE）、平均绝对百分比误差（MAPE）、平均绝对误差（MAE）和拟合度系数（EC）作为评价指标，分别定义如下：

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{34}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{35}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|(\hat{y}_i - y_i)\right| \tag{36}$$

$$EC = 1 - \frac{\sqrt{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}}{\sqrt{\sum_{i=1}^{N} y_i^2} + \sqrt{\sum_{i=1}^{N} \hat{y}_i^2}} \tag{37}$$

其中：N 为样本数量；$\hat{y}_i$ 和 y_i 分别表示数据的预测值和真实值。

上述评价指标中，MSE、MAPE 和 MAE 的取值越小，表明预测误差越小，相应模型的预测性能越好；EC 的取值越接近于 1，表明预测值与真实值之间的拟合程度越高，二者之间有更加相似的演化趋势。

4.4　预测结果对比分析

采用 MATLAB R2013a 进行交通流预测仿真实验，编程调用 LSSVMlab1.8 工具箱。表 4 给出了不同的智能优化算法寻优 LSSVM 模型进行交通流预测的参数寻优结果。图 3 至图 6 分别描述了基本 LSSVM 模型、FOA-LSSVM 模型、GSA-LSSVM 模型和 TCK-AGSA-LSSVM 模型在测试集上的预测效果对比，其中：i 表示时间索引；q 表示交通流量；m 表示真实交通流量数据；p 表示模型预测数据。上述模型的预测结果与真实数据的相对误差分布情况如图 7 所示，其中 R 表示不同模型的相对误差值。

表 4　不同预测模型参数寻优结果

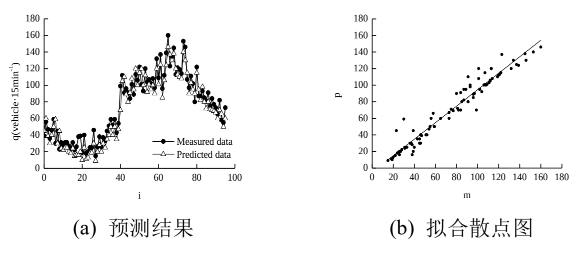
预测模型	C	σ
FOA-LSSVM	21.754	1.945
GSA-LSSVM	36.871	0.649
TCK-AGSA-LSSVM	39.527	1.386

从图 3（a）~图 6（a）可以看出，相比于其他三种预测模型，TCK-AGSA-LSSVM 模型的预测曲线与真实数据曲线的拟合效果最佳，具有更为相似的交通流变化趋势，适用性更强。

图 3（b）~图 6（b）给出了不同模型的预测结果与真实数据的拟合散点图，可见 TCK-AGSA-LSSVM 模型与其他模型相比，其预测结果与真实数据具有更小的分布偏差。

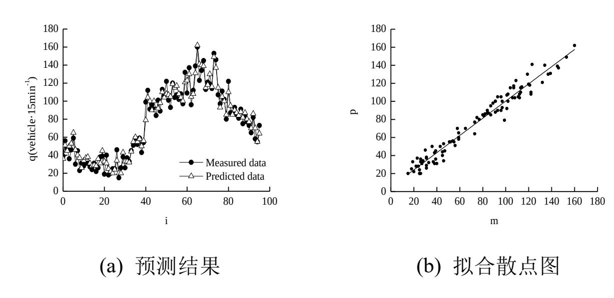
从图 7 的相对误差分布图可以看出，基于改进 TCK-AGSA 算法寻优 LSSVM 模型的预测相对误差波动范围最小，且大部分相对误差百分比较为稳定的维持在 5%以下，对于突变剧烈、较难预测的时间点（如 $i = 40$），相比于其他三种预测模型的预测精度也有了明显的改善，表明 TCK-AGSA-LSSVM 预测模型能够更加准确的反映交通流量演化趋势，更加贴近实际应用需求。

由表 5 可知，TCK-AGSA-LSSVM 模型的 MSE、MAPE 和 MAE 值均低于参比模型，其拟合度系数 EC 值均高于其他模型，表明该模型具有较小的预测误差，且拟合程度较高。



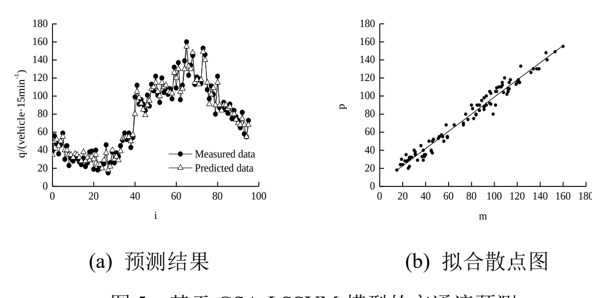
(a) 预测结果　　(b) 拟合散点图

图 3　基于 LSSVM 模型的交通流预测



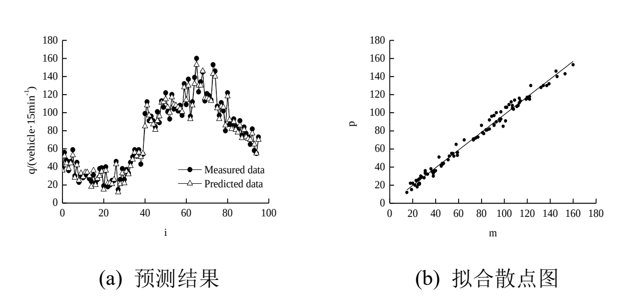
(a) 预测结果　　(b) 拟合散点图

图 4　基于 FOA-LSSVM 模型的交通流预测



(a) 预测结果　　(b) 拟合散点图

图 5　基于 GSA-LSSVM 模型的交通流预测



(a) 预测结果　　(b) 拟合散点图

图 6　基于 TCK-AGSA-LSSVM 模型的交通流预测

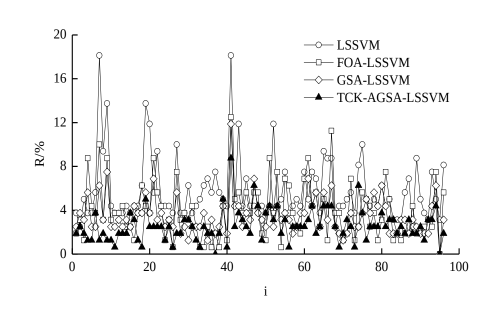


图 7　不同预测模型相对误差比较

表 5　不同预测模型指标值对比

预测模型	MSE	MAPE/%	MAE	EC
LSSVM	83.856	7.214	9.346	0.823
FOA-LSSVM	51.608	5.061	5.712	0.916
GSA-LSSVM	35.818	4.379	4.256	0.955
TCK-AGSA-LSSVM	**19.207**	**3.220**	**3.727**	**0.972**

综上所述，基于 TCK-AGSA 算法进行参数寻优的 LSSVM 模型预测性能良好，能够有效提高交通流量的预测精度和拟合质量。

5　结束语

本文首先提出一种改进的引力搜索算法 TCK-AGSA。利用 Tent 映射改进 Kbest 函数，使算法具有跳出局部最优的机制；

引入全局最优引导策略，使粒子加速朝向最优解移动；将进化度因子和聚合度因子引入速度更新权重系数，使算法具有较强的自适应能力，更好地平衡算法的全局探索能力和局部开发能力。通过在 12 个形态各异的基准测试函数上进行仿真实验，结果表明 TCK-AGSA 算法的寻优性能均优于基本 GSA 算法、GG-GSA 算法和 CKGSA 算法，从而验证了所提出算法在收敛精度和收敛速度上的优越性。

然后，利用改进算法 TCK-AGSA 对 LSSVM 参数组合进行寻优，构建 TCK-AGSA-LSSVM 交通流预测模型。选取 2016 年贵州省高速公路上的真实交通流数据进行仿真实验，结果表明，该模型的误差指标 MSE、MAPE、MAE 值均低于基本 LSSVM 模型、FOA-LSSVM 模型和 GSA-LSSVM 模型，且均等系数 EC 值高于其他三个预测模型，验证了 TCK-AGSA 寻优 LSSVM 的预测方法具有更好的预测精度、鲁棒性和泛化能力，其预测结果能够为实际交通诱导与交通管控提供有力的参考。

参考文献：

[1] Kalamaras I, Zamichos A, Salamanis A, *et al.* An interactive visual analytics platform for smart intelligent transportation systems management [J]. IEEE Trans on Intelligent Transportation Systems, 2018, 19 (2): 1-10.

[2] Rajabzadeh Y, Rezaie A H, Amindavar H. Short-term traffic flow prediction using time-varying Vasicek model [J]. Transportation Research Part C, 2017, 74: 168-181.

[3] Kumar S V, Vanajakshi L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data [J]. European Transport Research Review, 2015, 7 (3): 21.

[4] Kumar S V. Traffic flow prediction using Kalman filtering technique [J]. Procedia Engineering, 2017, 187: 582-587.

[5] Lv Yisheng, Duan Yanjie, Kang Wenwen, *et al.* Traffic flow prediction with big data: a deep learning approach [J]. IEEE Trans on Intelligent Transportation Systems, 2015, 16 (2): 865-873.

[6] Polson N G, Sokolov V O. Deep learning for short-term traffic flow prediction [J]. Transportation Research Part C: Emerging Technologies, 2017, 79: 1-17.

[7] Hu Wenbin, Yan Liping, Liu Kaizeng, *et al.* A short-term traffic flow forecasting method based on the hybrid PSO-SVR [J]. Neural Processing Letters, 2016, 43 (1): 155-172.

[8] 康军，段宗涛，唐蕾，等. 一种 LS-SVM 在线式短时交通流预测方法 [J/OL]. 计算机应用研究，2018, 35 (10) . http://www. arocmag. com/article/02-2018-10-022. html. (Kang Jun, Duan Zongtao, Tang Lei, *et al.* LS-SVM online short term traffic flow prediction method [J/OL]. Application Research of Computers, 2018, 35 (10) . http://www. arocmag. com/article/02-2018-10-022. html.)

[9] Shang Qiang, Lin Ciyun, Yang Zhaosheng, *et al.* Short-term traffic flow prediction model using particle swarm optimization-based combined kernel function-least squares support vector machine combined with chaos theory [J]. Advances in Mechanical Engineering, 2016, 8 (8): 1-12.

[10] Cong Yuliang, Wang Jianwei, Li Xiaolei. Traffic flow forecasting by a least squares support vector machine with a fruit fly optimization algorithm [J]. Procedia Engineering, 2016, 137: 59-68.

[11] Suykens J A K, Vandewalle J. Least squares support vector machine classifiers [J]. Neural Processing Letters, 1999. 9 (3): 293-300.

[12] 王鲜芳，张悦，王俊美. 基于 SAPSO-LSSVM 的蛋白质模型质量评估 [J]. 计算机应用研究，2017, 34 (5): 1346-1348+1378. (Wang Xianfang, Zhang Yue, Wang Junmei. Predicting protein model quality assessment by SAPSO-LSSVM [J]. Application Research of Computers, 2017, 34 (5): 1346-1348+1378.)

[13] 耿立艳，陈丽华. 基于 FOA 优化混合核 LSSVM 的铁路货运量预测 [J]. 计算机应用研究，2017, 34 (2): 409-412. (Geng Liyan, Chen Lihua. Forecast on railway traffic volume using mixed-kernel LSSVM optimized by FOA [J]. Application Research of Computers, 2017, 34 (2): 409-412.)

[14] Yuan Xiaohui, Chen Chen, Yuan Yanbin, *et al.* Short-term wind power prediction based on LSSVM–GSA model [J]. Energy Conversion & Management, 2015, 101: 393-401.

[15] Sun Wei, Sun Jingyi. Daily PM2. 5 concentration prediction based on principal component analysis and LSSVM optimized by cuckoo search algorithm [J]. Journal of Environmental Management, 2016, 188: 144-152.

[16] Liu Chao, Niu Peifeng, Li Guoqiang, *et al.* A hybrid heat rate forecasting model using optimized LSSVM based on improved GSA [J]. Neural Processing Letters, 2017, 45 (1): 299-318.

[17] Li Yanan, Yang Peng, Wang Huajun. Short-term wind speed forecasting based on improved ant colony algorithm for LSSVM [J]. Cluster Computing, 2018, 9: 1-7.

[18] Kumar L, Sripada S K, Sureka A, *et al.* Effective fault prediction model developed using least square support vector machine (LSSVM) [J]. Journal of Systems & Software, 2018, 137: 686-712.

[19] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm [J]. Information Sciences, 2009, 179 (13): 2232-2248.

[20] Li Chaoshun, Zhou Jianzhong. Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm [J]. Energy Conversion and Management, 2011, 52 (1): 374-381.

[21] Darzi S, Kiong T S, Islam M T, *et al.* A memory-based gravitational search algorithm for enhancing minimum variance distortionless response beamforming [J]. Applied Soft Computing, 2016, 47 (C): 103-118.

[22] Zhang Nan, Li Chaoshun, Li Ruhai, *et al.* A mixed-strategy based gravitational search algorithm for parameter identification of hydraulic turbine governing system [J]. Knowledge-Based Systems, 2016, 109: 218-237.

[23] 郭洁皓，高兴宝. 基于信息熵的混合引力搜索算法 [J]. 计算机应用研究，2016, 33 (05): 1319-1321, 134. (Guo Jiehao, Gao Xingbao. Hybrid gravitational search algorithm with information entropy [J]. Application Research of Computers, 2016, 33 (05): 1319-1321, 134.)

[24] Mittal H, Pal R, Kulhari A, *et al.* Chaotic Kbest gravitational search algorithm (CKGSA) [C]// Proc of the 9th International Conference on Contemporary Computing. Piscataway, NJ: IEEE Press, 2017: 1-6.

[25] Bohat V K, Arya K V. An effective gbest-guided gravitational search algorithm for real-parameter optimization and its application in training of feedforward neural networks [J]. Knowledge-Based Systems, 2018, 143: 192-207.